DSA lab Programs

M. Narasimha

AP19110010258

CSE-H

1. Write a program for insertion sort algorithm

Program:

```
#include <math.h>

#include <stdio.h>



void insertionSort(int arr[], int n)

{

   int i, key, j;

   for (i = 1; i < n; i++) {

      key = arr[i];

      j = i - 1;



      while (j >= 0 && arr[j] > key) {

         arr[j + 1] = arr[j];

         j = j - 1;

      }

      arr[j + 1] = key;

   }

}

void printArray(int arr[], int n)

{

   int i;

   for (i = 0; i < n; i++)
```

```
        printf("%d ", arr[i]);
    printf("\n");
}

int main()
{
    int arr[] = { 11, 5, 19, 4, 13 };
    int n = sizeof(arr) / sizeof(arr[0]);

    insertionSort(arr, n);
    printArray(arr, n);

    return 0;
}
```

Output:

4 5 11 13 19

2. Write a program for selection sort algorithm

Program:

```
#include <stdio.h>

void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void selectionSort(int arr[], int n)
{
```

```
    int i, j, min_idx;

    for (i = 0; i < n-1; i++)
    {

        min_idx = i;
        for (j = i+1; j < n; j++)
          if (arr[j] < arr[min_idx])
            min_idx = j;

        swap(&arr[min_idx], &arr[i]);
    }
}

void printArray(int arr[], int size)
{
    int i;
    for (i=0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}



int main()
{
    int arr[] = {15, 12, 11, 13, 14};
    int n = sizeof(arr)/sizeof(arr[0]);
    selectionSort(arr, n);
    printf("Sorted array: \n");
    printArray(arr, n);
    return 0;
```

```
}
```

Output:

sorted array

11 12 13 14 15

3. Write a program for bubble sort algorithm

Program:

```
#include <stdio.h>


void swap(int *xp, int *yp)
{
   int temp = *xp;
   *xp = *yp;
   *yp = temp;
}



void bubbleSort(int arr[], int n)
{
  int i, j;
  for (i = 0; i < n-1; i++)


     for (j = 0; j < n-i-1; j++)
        if (arr[j] > arr[j+1])
           swap(&arr[j], &arr[j+1]);
}



void printArray(int arr[], int size)
```

```
{
   int i;
   for (i=0; i < size; i++)
      printf("%d ", arr[i]);
   printf("n");
}



int main()
{
   int arr[] = {12, 34, 29, 19, 22, 11, 90};
   int n = sizeof(arr)/sizeof(arr[0]);
   bubbleSort(arr, n);
   printf("Sorted array: \n");
   printArray(arr, n);
   return 0;
}
```

Output

Sorted array

11 12 19 22 29 34 90

4. Write a program for merge sort algorithm

Program

```
#include<stdlib.h>
#include<stdio.h>



void merge(int arr[], int l, int m, int r)
{
   int i, j, k;
```

```
int n1 = m - l + 1;
int n2 =  r - m;


int L[n1], R[n2];


for (i = 0; i < n1; i++)
   L[i] = arr[l + i];
for (j = 0; j < n2; j++)
   R[j] = arr[m + 1+ j];


i = 0;
j = 0;
k = l;
while (i < n1 && j < n2)
{
   if (L[i] <= R[j])
   {
      arr[k] = L[i];
      i++;
   }
   else
   {
      arr[k] = R[j];
      j++;
   }
   k++;
}
```

```
    while (i < n1)
    {
        arr[k] = L[i];
        i++;
        k++;
    }


    while (j < n2)
    {
        arr[k] = R[j];
        j++;
        k++;
    }
}


void mergeSort(int arr[], int l, int r)
{
    if (l < r)
    {

        int m = l+(r-l)/2;


        mergeSort(arr, l, m);
        mergeSort(arr, m+1, r);

        merge(arr, l, m, r);
    }
```

```
}



void printArray(int A[], int size)
{
   int i;
   for (i=0; i < size; i++)
      printf("%d ", A[i]);
   printf("\n");
}



int main()
{
   int arr[] = {14, 11, 13, 5, 6, 8};
   int arr_size = sizeof(arr)/sizeof(arr[0]);



   mergeSort(arr, 0, arr_size - 1);


   printf("\nSorted array is \n");
   printArray(arr, arr_size);
   return 0;
}
```

Output:

sorted array is

5 6 8 11 13 14